

REMARKS

Applicants thank the Examiner for the careful consideration given the present application. The application has been carefully reviewed in light of the Office action, and amended as necessary to more clearly and particularly describe the subject matter which applicants regard as the invention.

This amendment is responsive to the Office action mailed February 5, 2008 in connection with the above-identified patent application. In that action, claims 1-9 and 15-42 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,793,979 to Lichtman, et al. (hereinafter "Lichtman") in view of U.S. Patent No. 6,016,500 to Waldo, et al. (hereinafter "Waldo").

Applicants maintain, however, that claims 1, 15, 24, 36, 39, and 40 and their respective dependent claims are in condition for allowance. Each of independent claims 1, 15, 24, 36, 39, and 40 has been amended previously to distinguish over the art cited by the Examiner. Further amendments to those claims are made herein to clarify the claims only and for purposes of advancing prosecution in this matter.

The Present Application:

The present application is directed to a method, system and article of manufacture for releasing resources. In the prior art, a garbage collector 40 such as may be provided in an automatic memory management module, frees user session objects in an about to expire user session thereby helping to minimize an accumulation of memory leaks. In the present application, however, a resource deallocation module (RDM) 50 executes prior to the execution of the traditional

garbage collection. The resource deallocation module frees obsolete references of the user session objects before the garbage collection frees the user session objects. When a user session is about to expire and before the garbage collector is invoked or executed, the resource deallocation module is executed to remove any obsolete resources. In that way, the obsolete references of the user session objects are freed (by the RDM) followed by the user session objects being freed/by the traditional garbage collector. The resource deallocation module 50 accesses the object graph of the user session, traverses the object graph, dereferences user session objects by applying methods to remove references by a set of rules for a given user session object, and also deallocates any resources owned by the user session object by applying methods to release resources by a set of rules for a given object. This helps to address the problem of memory leaks due to obsolete references.

As an example of this, independent **claim 1** recites a method of releasing resources of a user session operating in a software environment and includes an automatic memory management algorithm executed by a garbage collector. An impending execution of the automatic memory management algorithm by the garbage collector for removing user session objects created for the user session is detected by the RDM. When the impending execution of the garbage collector is detected and prior to the automatic memory management algorithm removing session objects created for the user session, the resource deallocation module accesses a session object of the user session and traverses an object graph and identifies one or more obsolete external resource references of the session object, wherein the one or more obsolete external resource references have not been

released by the session object, and releases the one or more obsolete external resource references by a set of rules for the session object, and repeats the accessing, identifying, and releasing from each session object of the user session. The accessing, identifying, and releasing is performed by resource deallocation module prior to invoking the garbage collector. After the obsolete references are removed by the RDM, the traditional garbage collector may be invoked for removing the user session objects created for the user session that are not referenced by a remaining object and which do not reference one or more external resources.

In general, therefore, the resource deallocation module RDM detects an impending execution of the automatic memory management algorithm by the garbage collector and is responsive thereto for accessing, identifying, and releasing the one or more obsolete external resource references prior to the garbage collector executing the automatic memory management algorithm.

"Detecting":

On pages 2-16 of the Office action, the Examiner essentially repeated arguments made earlier in the record. A small refinement in the Examiner's position was made, however, on page 17 of the Office Action whereat the Examiner cited to MPEP 2106.

In the previous Office Action, the Examiner cited to Lichtman at column 43, lines 11-14 in support of her position that Lichtman shows "detecting an impending execution of the automatic memory management algorithm." However, on page 4 of the previous Office Action, the Examiner stated in the record that "Lichtman does not disclose wherein the garbage collector for removing a session object created for the

user session, wherein said session object of the user session is not referenced by a remaining object and which does not reference one or more external resources."

In accordance with the above, therefore, applicants respectfully submitted that since Lichtman does not disclose a garbage collector as noted by the Examiner at the bottom of page 4 of the Office Action, Lichtman cannot possibly "detect an impending execution of the automatic memory management algorithm" by the garbage collector as set out in independent claim 1 of the present application. Waldo uses the expression "garbage collector" but the only suggestion to combine it with Lichtman comes from the pending claims. As argued previously by applicants, the actions of Lichtman's assignment element cited in the Office Action are in response to a "deallocate" instruction by a configuration manager. This is not equivalent to "detecting an impending execution of the automatic memory management algorithm" as set out in the claims. In Lichtman, there does not appear to be any step of detecting an impending execution of another algorithm so that, prior to executing the automatic memory management algorithm by the garbage collector, the resource deallocation module can access, identify, and release one or more obsolete external resource references before the automatic memory management algorithm of the garbage collector removes the session objects created for the user session.

In the Office action of February 5, 2008, the Examiner maintained and refined her position that the combination of Waldo and Lichtman teaches the limitation of "detecting and impending execution of the automatic memory management algorithm" by the garbage collector. According to the Examiner, specifically, the Lichtman teaching in column 43, lines 11-14 meets applicant's teaching of "detecting

an impending execution of the automatic memory management algorithm" and Waldo goes on to teach this garbage collector in Figure 4, item 420 and column 8, lines 20-35. Hence, according to the Examiner, the combination of the two references teaches applicants' claim limitations.

Applicants respond as follows. First, in Lichtman at column 43, lines 11-14, it is indicated that "in response to a 'deallocate' instruction by the configuration manager 158, the assignment element deletes the resource assignments in the assigned resource table 304 for the designated device 20. This permits those resource elements to be assigned for use by other devices 20." Thus, using the Examiner's reasoning, the assignment element in Lichtman "detects an impending execution of the automatic memory management algorithm" because it receives a "deallocate" instruction by the configuration manager 158. Essentially, the Examiner has taken what amounts to this position on page 17 of the Office action in the clarification portion of the previous rejection. In addition, on page 2 of the Office action, the Examiner took the position that column 43, lines 11-14 of Lichtman is equivalent to "using the resource deallocation module detecting an impending execution of the automatic memory management algorithm. Specifically, the Examiner considered the assignment element deleting the resource assignments in the assigned resource table in response to a 'deallocate' instruction by the configuration manager to be equivalent to applicant's claimed "using the resource deallocation module, detecting an impending execution of the automatic memory management algorithm."

While applicants do not concede that a direct "deallocate" instruction from a configuration manager 158 to an assignment element causing the assignment

element to delete the resource assignments in the assigned resource table for the designated device to be the same or equivalent to "detecting an impending execution of the automatic memory management algorithm" as set out in the claims, it appears that the assignment element in Lichtman is directly responsive to instructions from the configuration manager to delete resource assignments. Based on this line of reasoning then, the Examiner cites to Waldo for teaching a garbage collector for removing a session object created for a user session and took the position that it would have been obvious to one of ordinary skill in the art to incorporate the Waldo teachings into the Lichtman system.

Therefore, based upon the above, the Examiner is suggesting incorporating the "garbage collector" teachings of Waldo into the Lichtman system so that Waldo's garbage collector can be executed based upon a "deallocate" instruction from the configuration manager to delete resource assignments in an assigned resource table for a designated device to perform a garbage collection function.

Applicants respectfully submit that, even if the combination as suggested by the Examiner was possible, it still would not teach, suggest, or fairly disclose the features of the independent claims in the present application.

In the Examiner's combination, the Waldo "garbage collector" would be responsive to a "deallocate" instruction by a configuration manager. However, in the present application, the garbage collection is performed by the garbage collector 40 only after specific operations are conducted by the resource deallocation module (RDM) 50.

As described in the specification such as, for example, on page 5, the garbage collection algorithm or algorithms determine whether each user session

object is still needed and, removes user session objects which are no longer needed.

However, the typical garbage collector 40 can only remove user session objects which are not referenced by any other user session objects, and which do not include references to external resources. This being the case, the garbage collector 40 may fail to collect certain user session objects such as, for example, certain user session objects which fail to properly dereference one or more system resource references after the user session object is finished accessing the corresponding system resource. These obsolete references can include, for example, file handles, database connections, sockets, threads, or the like. In the prior art system, these obsolete resources are detected by the garbage collector 40 and misinterpreted as indicating that the user session object is still needed. Therefore, the garbage collector 40 acting alone as in the prior art fails to collect the user session object.

As set out on page 8 of the present application, a resource deallocation module (RDM) 50 is executed prior to the garbage collection to deallocate any remaining allocated external resources of the about-to-expire user session. The RDM in accordance with the present application effectively addresses memory leaks due to obsolete references. The resource deallocation module 50 accesses the object graph of the user session, traverses the object graph, dereferences user session objects by applying methods to remove references by a set of rules for a given user session object, and also deallocates any resources owned by the user session object by applying methods to release resources by a set of rules for a given object.

As further set out on page 10, beginning at line 21 of the present application, after the resource deallocation module 50 has gone through the enumerated objects

of the user session and completed the task of releasing any remaining references within each object, the garbage collector 40 is invoked to perform the usual garbage collection for the about-to-expire user session. Since any obsolete references of the user session objects have been removed by the resource deallocation module 50, the garbage collector frees the user objects without leaving memory leaks in the form of unfreed user session objects left over due to improper hanging references.

Claims 1-9 and 33-35 are in Condition for Allowance:

Applications have tendered an amendment to independent claim 1 above for clarification purposes and to help advance prosecution in this application. It is respectfully submitted that independent claim 1 distinguishes over the art of record.

In particular, claim 1 recites a method in a resource deallocation module of releasing resources of a user session operating in a software environment that includes an automatic memory management algorithm executed by a garbage collector. In broad terms, the method comprises detecting an impending execution of an automatic memory management algorithm by a garbage collector, removing any obsolete references of the user session objects prior to executing the automatic memory management algorithm by the garbage collector for removing the user session objects created for the user sessions that are not referenced by a remaining object and which do not reference one or more external resources.

In the Examiner's combination of Lichtman with Waldo, the Waldo "garbage collector" is responsive to a "deallocate" instruction from the configuration manager for deleting resource assignments in an assigned resource table for a designated device. In independent claim 1, however, first, the resource deallocation module

detects the impending execution of the automatic memory management algorithm by the garbage collector for removing user session objects created for the user sessions that are not referenced by a remaining object and which do not reference one or more external resources. Responsive to the detecting and prior to executing the automatic memory management algorithm by the garbage collector, the resource deallocation module removes any obsolete references of the user session objects by traversing an object graph, identifying one or more obsolete external resource references, releasing the one or more obsolete external resource references, and repeating the accessing, identifying, and releasing for each user session object of the user session objects, thereby removing any obsolete references of the user session objects. Then, after the removing of the obsolete references by the resource deallocation module, the automatic memory management algorithm may be invoked for removing the user session objects created for the user sessions that are not referenced by a remaining object and which do not reference one or more external resources.

In dependent claim 6, the automatic memory management algorithm is executed by the garbage collector after removing the obsolete references of the user session objects by the resource deallocation module to remove the user session objects created for the user session.

It is respectfully submitted that the Lichtman and Waldo references, alone or in combination, do not teach, suggest, or fairly disclosure the two "phase" approach recited in independent claim 1 wherein a resource deallocation module 50 first removes any obsolete references of the user session objects, essentially "clearing

the way" for an invocation of an automatic memory management algorithm by a garbage collector.

For at least the above reasons, it is respectfully submitted that independent claim 1 and claims 2-9 and 33-35 dependent therefrom are patentably distinct and unobvious over the references of record.

Claims 15-32 and 36-42 are in Condition for Allowance:

In addition to the above, applicants have tendered similar amendments to clarify each of independent claims 15, 24, 36, 39 and 40 above wherein the claim language sets out that the resource deallocation module is responsive to the an impending execution of an automatic memory management algorithm by a garbage collector and, further, that the releasing of the resources by the resource deallocation module is performed prior to executing the automatic memory management algorithm by the garbage collector.

For at least the above reasons, it is respectfully submitted that independent claims 15, 24, 36, 39, and 40 and their respective dependent claims are patentably distinct and unobvious over the references of record.

In light of the foregoing, it is respectfully submitted that the present application is in a condition for allowance and notice to that effect is hereby requested.

If it is determined that the application is not in a condition for allowance, the Examiner is invited to initiate a telephone interview with the undersigned attorney to expedite prosecution of the present application.

If there are any additional fees resulting from this communication, please charge same to our Deposit Account No. 09-0460, our Order No. SVL920030044US1.

Respectfully submitted,

RANKIN, HILL & CLARK LLP

By /Michael E. Hudzinski/
Michael E. Hudzinski, Reg. No. 34185

38210 Glenn Avenue
Willoughby, Ohio 44094-7808
(216) 566-9700